Today's

1. Pattern matching → KMP

a	x
b	x
c	x
d	x
e	x

2. Given an array and we have [l, r] in [0, n-1]
find the sum of all elements in the range [l, r]
in constant time. Array is not mutated.

A = [1, 2, 3, 7, 3, 5, 1, 4, 6, 2]

① → O(n) per query

If you are allowed

prefix sum array

$A_{pre}[i] = A_{pre}[i-1] + A[i]$

$A_{pre}[r] - A_{pre}[l-1] + A[l]$

$A_{pre}[r] - A_{pre}[l-1]$

O(log n) per query

1	3	6	10	13	18	19	23	29	31

O(1) per query

$O(n) \longrightarrow O(1)$

Q [100, 10², 10³, 10⁴, 10⁵]

100	10²	10³	10⁴	10⁵

Hash Functions

① Hash $0 \le h(x) \le M < n$

[10, 20, 30, 32] M = 10

$h(10) = 10 \bmod 10 = 0$

$h(20) = 20 \bmod 10 = 0$

$h(30) = 30 \bmod 10 = 0$

$h(32) = 32 \bmod 10 = 2$

Collision

M = 1 → all keys are colliding → collision

② Chaining

A = {7, 4, 5, 9, 8, 14, 2, 6, 8}

M = 5

key	value
0	
1	
2	
3	
4	

$h(x) = x \bmod 5$

→ 3/5

Insert M = 5

[5, 9, 0, 4, 7, 10]

0	0 → 5 → 10
1	
2	7 → 2
3	
4	9 → 4

$h(x) = x \bmod 5$

Best TC → O(1)

Worst TC → O(n)

$\frac{n}{5}$

$h \rightarrow \log n$

Hashing stores the information in an array form by key-value pairs

Ex: hashing "Ankit" value is 20

(key, value)

	C++	Java
Unordered Set	set	HashSet
Unordered Map	map	Hash Map

Unordered Set TC: O(1) Avg, O(n) Worst

Unordered Map TC: O(1) Avg, O(n) Worst

Set → O(log n)

Map → O(log n)

Set

1	0
2	1
3	2
4	3
5	4

Hash Set

Set → Hash → Array

Map

	value
1	0
2	1
3	2
4	3
5	4

Hashing

map<int, int> m;

int n, k, ans;

cin >> n >> k;

for (int i = 0; i < n; i++) {

int x;

cin >> x;

m[x] = i;

int y = m[k - x];

if (y != i && y != -1) {

ans = max(ans, i - m[k-x])

}

m[x] = i;

}

TC: O(n)

SC: O(n)

3. Given an array, find largest area subarray

Sum 0

[1, -2, 2, -1, 0, 4]

① → vl: 6×4

1	-1	1	1	1	5
0	1	0	1	2	3

0

$P[i] = \sum_{j=0}^{i} A[j]$

i → 0, j

$P[j] = A[0] + A[1] + \dots A[i-1] + \dots A[j]$ − $P[i-1]$

$P[j] = P[i-1]$

Finding 2 point (i, j) s.t. P[j] == P[i-1] & (i-j) is maximum

1	-2	2	-1	0	4
0	1	-1	0	-1	3

1 → ③ X

3	-2	2
0	-1	0

1	2	3	-2	-4
0	1	0	1	0

0 → -1	
1 → 0	
3 → 1	
6 → 2	
4 → 3	

TC: O(n)

Space: O(n)

Q. "abkeeaade"

"ckbaaade"

c<b<e<a<d<e

Sol. Naive

TC: O(|S|·|d|)

Space: O(1)

O(1)

O(1)

Alphabetic Hashing

1	3
a	b

a<b

c → 0
b → 1
f → 2
a → 3
d → 4
e → 5

Comparator Function

sort (A.begin(), A.end(), cmp);

bool cmp(int x, int y) {

return m[x] < m[y]

}

TC: O(n log n)

Q. A → [100, 4, 3, 6, 10, 12, 11, 5, 101]

Find the longest seq. of numerically consecutive elements

[3, 4, 5, 6] → Ans = 4

[100, 101]

[10, 11, 12]

$0 \le A[i] \le 10^9$

$N \le 10^6$

Brute Force

Sort the array → O(n log n)

[100 4 3 6 10 12 11 5 101]

a+1 a+2 a+3

Just increase

count

3

unordered set

TC: O(2×N)

O(n)

Space: O(n)

Queues

Que 3 types of Queries

① X : int ⇒ add one occurrence of X

② X : int ⇒ remove one occurrence of X

③ Binary string

Online queries

<1, 256>, <1, 323>, <1, 121>, <2, 323>

<3, 101>

256	
323	
121	
120	

odd, even

1 → odd

0 → even

odd even odd